Important Method for Detecting and Tracking Based on Color

Zainab OUFQIR, Abdellatif EL ABDERRAHMANI, Khalid SATORI

***Abstract*: Our work presents a new robust approach that detects an object based on its color in a real 3D scene and in real time, this detected object will be augmented by additional information and its position in the scene will allow us to calculate the camera pose in order to correctly align the virtual object in the real world. The calculation of the camera position is the most studied problem in augmented reality, the solution of this problem is to provide the position of four 3D points in a real-world reference and their 2D position of their projection in the image sequence acquired in real time. For good detection, the color mode conversion of the scene from RGB to HSV allows multiple colors to be extracted and this color space is less sensitive to shadows and variance of brightness. A rectangle is automatically drawn around the detected object and surrounds the minimum surface area of its contour, this rectangle expresses the location of the detected object in the scene and the position of the four corners of this rectangle will allow us to follow the object in the scene so we can estimate the position of the camera. The result shows the quality, simplicity and speed of our approach compared to the methods mentioned in the related work section.**

***Keywords*: augmented reality, color, object detection, real time.**

I. INTRODUCTION

Augmented reality enriches the real world with additional information in a defined situation or in a specific environment via a smartphone or glasses [1] [2]. It allows to combine one or more virtual objects with the real world, in other words they coexist together [3]. The user stays connected to the real world and is able to visualize a scene augmented by texts, images or 3D objects and achieves visual consistency. Unlike virtual reality, where the user is totally immersed in a virtual world. Devices (Smartphone, tablet...) are becoming more and more powerful and efficient, they provide the necessary tools (camera, screen) to create an application to insert an object into a real 3D scene [4].

Revised Manuscript Received on June 12, 2019.

First Author name, His Department Name, University/ College/ Organization Name, City Name, Country Name.

Second Author name, His Department Name, University/ College/ Organization Name, City Name, Country Name.

Third Author name, His Department Name, University/ College/ Organization Name, City Name, Country Name.

The difficulty encountered in augmented reality is the understanding of the geometric structure of a scene visualized through a camera, this means detecting and describing visual characteristics of the image to find the position of the camera in the real world reference frame, which will allow to correctly align the object inserted in the real world [5].This problem has been the subject of several scientific studies in the field of computer vision because the object detection process is the most important process in the calculation of the pose. We will present a new approach that consists in detecting an object based on its color. In the first chapter, we will describe the different methods existing in augmented reality to detect an object or characteristics of a 3D scene, then we will detail our approach by ensuring its robustness and speed.

II. RELATED WORK

In computer vision, object detection is a method of detecting the presence of an instance or class of objects in a digital image. Computer vision aims to understand the content of an image, this understanding involves in particular the recognition of objects in images and the understanding of the interaction relationships between objects and the scene, which has attracted the interest of the scientific community. A digital image is a projection of a real scene in an image plane, this projection is made through a sensor that is the basic component of a camera. A sensor is a sensitive surface that, thanks to its millions of pixels, will receive light from the lens of a camera. It will then transform this light into electrical signals that will be converted into digital images using a processor.The pinhole model [6] models a camera by a perspective projection, it allows to transform all the 3D points of a real scene into a 2D image plane, it leads to a geometric model that is mathematically rich. It is generally used more in currency cameras (smartphones, tablets...) which are configured according to this model.

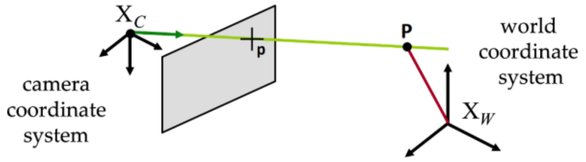

Fig 1: Representation of pinhole model

This model describes by a mathematical formula the relationship between a 3D

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

point of the real world and its projection in the image plane:

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = K\left[R \;\; t\right]\begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} \quad (1)$$

K represents the matrix of intrinsic camera parameters, (R, t) represents the matrix of extrinsic camera parameters. Calculating the camera position is equivalent to finding the rotation R and translation t with regard to a reference in the real world, which allows a virtual object to be correctly aligned in a scene. This process begins by detecting 2D points of a particular object and tracking them in real time in the images acquired in real time [7]. The extraction of primitives is one of the methods used in augmented reality, these primitives can be in the form of a line [8], point [9], contour [10]. Corner extraction (example Harris [11]) has become the most widely used technique, due to its good performance in terms of repeatability and processing time.The extraction of key points (example SIFT [12], SURF [13], FAST [14]) is also the most widely used in computer vision. It offers robust detection and is characterized by a unique descriptor that allows the object to be located in the different incoming images.Contours are also rich indices and are widely used as a pre-processing step for detecting objects to find the boundaries of regions. An object can be located from all the pixels of its contour, they can be detected thanks to the filters of types Sobel [15], Prewitt [16] , Roberts [17] ou Canny [18]. There are algorithms that detect an object based on their 3D model [19], which requires the use of a camera that analyzes the scene in 3 dimensions, the process of detecting key points on different points of view is done offline [20] which presents a limitation for augmented reality.Shape detection recognizes an object in an unknown environment based on its geometric shape. The process consists of reducing the noise in the image, detecting the contours and comparing the result to a database [21]. If an object model is available, object detection is equivalent to matching the different primitives extracted from the image.The presence of a marker in the scene reduces the cost of calculating by knowing its geometric shape beforehand. The marker-based method is a method that detects a marker as a 2D image added in the real world, the markers are used as a physical reference point in a scene and they are easily identifiable by knowing their geometry. This method offers a robust result and allows the virtual object to be inserted precisely. The only constraint is that the marker must always stay visible in the observed scene and their dimensions must be known. This method is popularized by ARToolkit [22], it is used by many augmented reality applications, it consists in placing a black and white marker of a black square border surrounding a model in a scene. Detection of the four corners of the square is sufficient to calculate the camera position.The use of these methods is expensive in computation time and constitutes a constraint for augmented reality where everything happens in real time, we will present our approach and ensure its effectiveness.

III. OUR APPROACH

Our method describes a robust technique to locate an object in a real 3D scene in real time visualized by the camera of a device (smartphone, tablet...). The idea of this research is to detect the object based on its color in HSV mode, this color mode has a variety of advantages that allows good color detection using their Hue, Saturation and Value intervals [23]. Extracting the color of the object facilitates the process of detecting contours in a binarized image, then we draw a rectangle around the minimum surface of the extracted contour to express the correct detection. The purpose of our method is to be able to locate an object easily and quickly in a scene to augment it and to exploit it in the process of calculating the camera pose for augmented reality based on the position of the corners of this rectangle. We used the OpenCV library [24] for the implementation of our method, it is a free graphic library, specialized in real-time image processing. The following figure shows the algorithm implemented for our method.

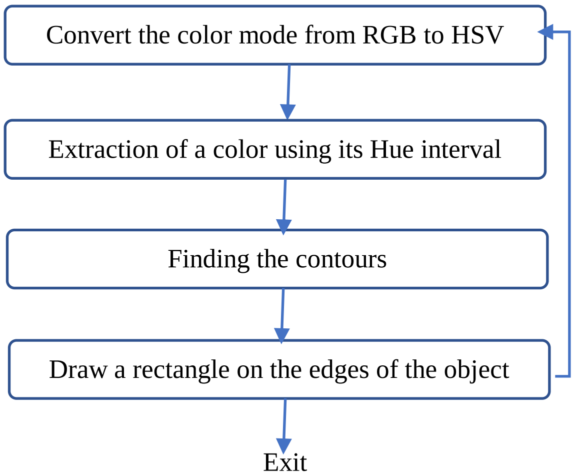

Fig 2: Diagram of the processing of a 3D scene (image sequence)

A. *Convert the color mode*

The method used to extract the colors is HSV (Hue Saturation Value), it is more intuitive to identify and describe the colors, it offers a wide possibility to extract and know several colors by knowing their intervals H, S and V. This color space is less sensitive to shadows and variance in brightness [23].

The HSV model consists of decomposing the color according to physiological criteria:

- **Hue**, is the pure shape of a color, it varies between 0 and 360°
- **Saturation**, describes the purity of the color and the intensity of the coloration
- **Value**, indicates the quantity of light of the color, it expresses the impression of clarity, brightness of the color

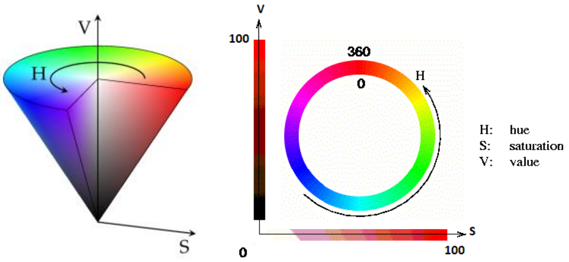

Fig 3: Color HSV

The first step of our method is to convert the color mode of the scene from RGB to HSV. The calculation of each Hue Saturation Value is done as follows:

$$V = \max(R,G,B) \quad (2)$$

$$S = \begin{cases} V - \dfrac{\min(R,G,B)}{V} & if\ V \neq 0 \\ 0 & else \end{cases} \quad (3)$$

$$H = \begin{cases} 60\left(\dfrac{G-B}{V-\min(R,G,B)}\right) & \text{if } V = R \\ 120 + 60\left(\dfrac{B-R}{V-\min(R,G,B)}\right) & \text{if } V = G \\ 240 + 60\left(\dfrac{R-G}{V-\min(R,G,B)}\right) & \text{if } V = B \end{cases} \quad (4)$$

B. *Color extraction*

To extract a color, we use the dedicated Hue interval for each one. Concerning saturation and brightness they vary between 10 and 255 which improves color detection in low light conditions. The following figure shows the different Hue intervals of each color, Hue is the most important criteria and one of the main properties of a color and it is invariant in terms of brightness, it is coded according to the angle that corresponds to it on the color circle:

- 0° ou 360° : rouge ;
- 60° : jaune ;
- 120° : vert ;
- 180° : cyan ;
- 240°: bleu;
- 300°: magenta.

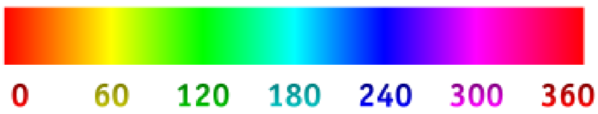

Fig 4: Hue interval

C. *Contour detection*

The contours are a curve connecting all continuous points with the same color or intensity. For better accuracy, it is preferable to use binary images. Finding contours is like finding a white object on a black background. For our method, extracting the desired color automatically produces a binary image containing the object detected in white and the rest of the scene in black, which facilitates the contour extraction process. The SUZUKI method [25] allows to extract the topological structure of a given binary image that includes contours that are a useful tool for shape analysis and object detection and recognition. Contours are determined as the boundaries between the black and white points where each contour determines a set of linked 2D points. The following figure shows an image of a real 3D scene, extracting the red color allows to display the detail of the flower outlines in the binarized image.

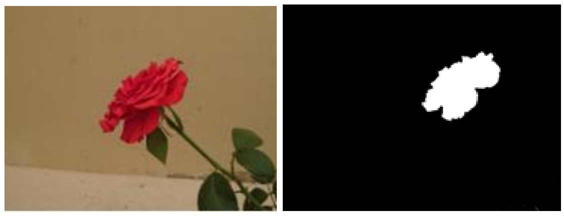

Fig 5: Image of a binarized scene displaying details of a red flower

D. *Detect and track the object*

Thanks to the contours detected in the binarized image, it becomes easy to locate our object in the scene. To draw the rectangle around the object, we will implement the Toussaint algorithm [26]. It surrounds the minimum surface of the contour, having a finite set of 2D points, we can calculate the minimum surface rectangle that contains the points of the polygon that contains all these points. Let us consider a set of points that belong to a contour, the points p_i, p_j, p_k, p_l are selected as candidates based on the minimum and maximum x and y coordinates. $L_s(p_i, p_j, p_k, p_l)$ represents a set of parallel straight lines that pass through each point, these lines are rotated to build a set of calipers with an angle, the vertices of the corners of the rectangle can be calculated from the coordinates when the optimal area of the rectangle is determined [27].

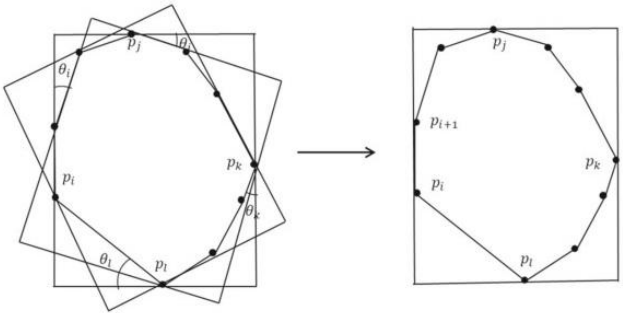

Fig 6: The optimal rectangle

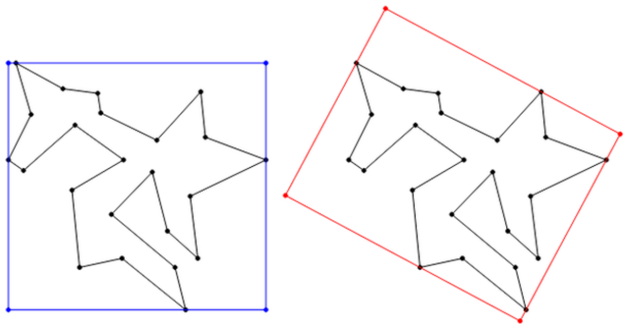

Fig 7: rectangle drawn around a contour

One of the specific features of this rectangle is that it is possible to know the position of its four corners in the image and follow them in real time, which is the number of points

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

necessary to calculate the position of the camera to insert a virtual object in the center of the rectangle and increase the detected object with additional information.

IV. EXPERIMENTATION

To study the performance of our method, we tested in different scenes to build its robustness. For each scene we applied the process described in our method. We tested on objects with different colors, using their Hue intervals and by varying saturation and brightness between 10 and 255 we were able to automatically detect different objects in the scene. The following figures show the results before and after the location of different objects in simple and complex real scenes, the object is located in real time through the camera of a device (smartphone, tablet...) using a green rectangle draw on its edges. The application is made with the android studio editor [28] using the OpenCV library [29] dedicated to image processing.

Fig 7: detection of a beige egg in a simple scene

Fig 8: detection of a turquoise thermos in a simple scene

Fig 9: detection of a red flower in a complex scene

Fig 10: detection of a brown door in a complex scene

V. INTERPRETATION

The results show the effectiveness of our method, the objects are recognized through their colors by specifying their Hue intervals and framed by a rectangle to express that the object is correctly detected. If several objects of the same color are present, the object with the largest number of grouped pixels will be surrounded. The rectangle surrounds the minimum contour area and provides the position of these four points as shown in figure 12, which allow the camera position to be calculated and correctly align a virtual object with the detected object. The robustness of this detection, its speed and its simplicity compared to the methods mentioned in the related work part, shows its performance for augmented reality where everything happens in real time. Thus we were able to complete the first step in the process of calculating the camera pose which allows us to detect and track an object in a scene.

Fig 11: Scene before detection

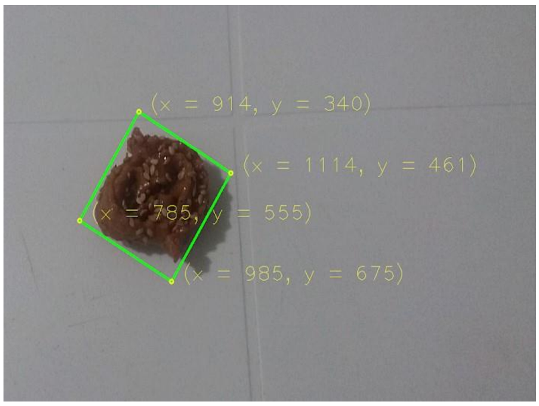

Fig 12: Scene after detection

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. CONCLUSION

This article presents a robust method for detecting and tracking an object in the real world in real time based on its color. The idea is to locate the object through its color in a real 3D scene using a rectangle drawn around the minimal surface of its contour. This rapid detection and knowledge of the position of the corners of this rectangle in the various images acquired in real time will facilitate the process of calculating the camera position for augmented reality.

REFERENCES

1. R. T. Azuma, « A Survey of Augmented Reality », p. 48.
2. R. Poelman, « Systems Engineering Section, Delft University of Technology Jaffalaan 5, 2628BX Delft, The Netherlands », p. 22.
3. H. Shoaib et S. W. Jaffry, « A Survey of Augmented Reality », p. 10.
4. M. de Sa, E. F. Churchill, et K. Isbister, « Mobile Augmented Reality: Design Issues and Opportunities », in Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, New York, NY, USA, 2011, p. 749–752.
5. D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, et M. Tuceryan, « Real-time vision-based camera tracking for augmented reality applications », 1997, p. 87-94.
6. K, Miyamoto, « Fish eye lens. J. Opt. Soc. Am. » .
7. V. Lepetit, J. Pilet, et P. Fua, « Point matching as a classification problem for fast and robust object pose estimation », in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Washington, DC, USA, 2004, vol. 2, p. 244-250.
8. R. DERICHE, O. FAUGERAS, et I. Sophia-Antipolis, « Tracking line segments », p. 10.
9. Daniel F. Dementhon et Larry S. Davis, « Model-based object pose in 25 lines of code ». .
10. A. I. Comport, E. Marchand, et F. Chaumette, « Robust model-based tracking for robot vision », 2004, vol. 1, p. 692-697.
11. C. Harris et M. Stephens, « A combined corner and edge detector », in In Proc. Fourth Alvey Vision Conference, 1988, p. 147–152.
12. D. G. Lowe, « Object recognition from local scale-invariant features », 1999, p. 1150-1157 vol.2.
13. H. Bay, T. Tuytelaars, et L. V. Gool, « SURF: Speeded Up Robust Features », p. 14.
14. E. Rosten et T. Drummond, « Fusing points and lines for high performance tracking », 2005, p. 1508-1515 Vol. 2.
15. I. Sobel, « Neighborhood coding of binary images for fast contour following and general binary array processing », Computer Graphics and Image Processing, vol. 8, no 1, p. 127-135, août 1978.
16. J. M. Prewitt et J. L. Prewitt-Freilino, « Object Enhancement and Extraction », janv. 1970.
17. L. G. Roberts, « Machine perception of three-dimensional solids », Thesis, Massachusetts Institute of Technology, 1963.
18. J. Canny, « A computational approach to edge detection », Ieee Transactions on Pattern Analysis and Machine InTELligence, 1986.
19. A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, et C. Rother, « 6-DOF Model Based Tracking via Object Coordinate Regression », in Computer Vision -- ACCV 2014, vol. 9006, D. Cremers, I. Reid, H. Saito, et M.-H. Yang, Éd. Cham: Springer International Publishing, 2015, p. 384-399.
20. W. Kehl et T. Munich, « Hashmod: A Hashing Method for Scalable 3D Object Detection », p. 15.
21. R.BARBER, M.FLICKNER, J.HAFNER, W. NIBLACK, et D.PETKOVIC, « Efficient and effective Querying by Image Content ».
22. « ARToolKit Home Page ». https://www.hitl.washington.edu/artoolkit/.
23. A. Bellarbi, S. Benbelkacem, N. Zenati-Henda, et M. Belhocine, « Hand gesture interaction using color-based method for tabletop interfaces », 2011, p. 1-6.
24. K. Pulli, A. Baksheev, K. Kornyakov, et V. Eruhimov, « Real-time computer vision with OpenCV », Communications of the ACM, vol. 55, no 6, p. 61, juin 2012.
25. S. SUZUKI, « Topological Structural Analysis of Digitized Binary Images by Border Following », p. 15.
26. G. Toussaint, « Solving Geometric Problems with the Rotating Calipers », p. 8.
27. Md. M. Rahman, K. Ishii, et N. Noguchi, « Optimum harvesting area of convex and concave polygon field for path planning of robot combine harvester », Intelligent Service Robotics, vol. 12, no 2, p. 167-179, avr. 2019.
28. « Download Android Studio and SDK tools ». https://developer.android.com/studio.
29. « OpenCV ». https://opencv.org/.

AUTHORS PROFILE

Zainab OUFQIR PhD student at university sidi mohamed ben abdellah, LIIAN Department of Mathematics and informatic Faculty of Sciences Dhar-Mahraz P.O.Box 1796 Atlas Fes, 30000, Morocco. Email address zayna.oufkir@gmail.com

Abdellatif EL ABDERRAHMANI professor at university Abdelmalek Essaâdi, LSTA,Departement of computer science Larache Poly disciplinary School LARACHE,Morocco. Email address elabderrahmani@yahoo.fr

Khalid SATORI professor at university sidi mohamed ben abdellah, LIIAN Department of Mathematics and informatic Faculty of Sciences Dhar-Mahraz P.O.Box 1796 Atlas Fes, 30000, Morocco. Email address khalidsatori@gmail.com

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication